Shared Autonomy Systems with Stochastic Operator Models

Clarissa Costen , **Marc Rigter** , **Bruno Lacerda** and **Nick Hawes**
Oxford Robotics Institute, University of Oxford, UK
{clarissa, mrigter, bruno, nickh}@robots.ox.ac.uk

Abstract

We consider shared autonomy systems where multiple operators (AI and human), can interact with the environment, e.g. by controlling a robot. The decision problem for the shared autonomy system is to select which operator takes control at each timestep, such that a reward specifying the intended system behaviour is maximised. The performance of the human operator is influenced by unobserved factors, such as fatigue or skill level. Therefore, the system must reason over stochastic models of operator performance. We present a framework for stochastic operators in shared autonomy systems (SO-SAS), where we represent operators using rich, partially observable models. We formalise SO-SAS as a mixed-observability Markov decision process, where environment states are fully observable and internal operator states are hidden. We test SO-SAS on a simulated domain and a computer game, empirically showing it results in better performance compared to traditional formulations of shared autonomy systems.

1 Introduction

As automation and robots become prevalent in our lives, designing systems that enable co-operation between humans and autonomy becomes necessary. In this work, we are interested in a shared autonomy (SA) setting, where a set of *operators* interact with the environment by taking control of a system (such as a robot) in order to contribute towards achieving a common goal. In the most general case the set of available operators may include both human teleoperators and autonomous controllers. An SA framework is responsible for deciding which operator should be in charge in order to manage progress towards the goal.

To enable this decision making, current SA approaches tend to use simplified models of their operators. For example, in Basich *et al.*, [2020] an autonomous controller drives a car until safety is compromised, at which point the human operator must take over. The authors make the assumption that the human driver is always perfect, therefore the SA system can always switch to the human operator to maintain safety. Cubuktepe *et al.*, [2021] consider a scenario where a human operator and an autonomous system share control of a wheelchair. The human in this setting is not fully aware of the risks present in the environment, but the autonomous system is assumed to have full visibility of the risks, and thus is able to always take appropriate action choices. Their model of the human operator makes the simplifying assumption that human performance does not evolve over time, e.g. due to learning or tiredness. This static human operator assumption is relaxed in Feng *et al.*, [2016], who present a SA system in which the human performance level might change due to underlying factors (fatigue). However, their model is simplified by the assumption that the onset of fatigue is deterministic, with the human transitioning from an alert to a tired state after a fixed number of timesteps. As each person has an individual level of tolerance to fatigue, it is unrealistic to assume that all operators are fatigued after the same number of steps.

In this paper, we relax the simplifying assumptions described above by introducing *performance profiles* that model the behaviour of the operators available to an SA system. These performance profiles depend on a set of partially observable state factors that evolve stochastically as the SA system acts. This leads us to the stochastic operators in shared autonomy systems (SO-SAS) model, which maintains a belief over the performance profile state for each operator. The observations gained from the interactions of the different operators with the environment are used to update the belief on which performance profile state each operator is in. To avoid the need for defining complicated observation functions, belief updates in SO-SAS are performed by observing environment state transitions.

Our main contribution is the SO-SAS framework, a novel formalisation of an SA system as a mixed-observability Markov decision process (MOMDP). To the best of our knowledge, this is the first SA framework that considers operator performance to be stochastic, dynamic and not directly observable by the SA decision-making framework. We conduct an empirical evaluation of SO-SAS on a simulated domain, and on a computer game, with comparison to existing modelling approaches. We show that our formulation results in better performance in comparison to using the simpler models (e.g. assuming a perfect human or static performance) typically used in previous work.

2 Related Work

When an autonomous system works in the same space as humans [Wu *et al.*, 2017], or shares control over a robot with a human [Feng *et al.*, 2016], they must be able to reason over potential human interaction. These human-aware autonomous systems typically determine the next action by attempting to maximise the rewards collected over the future timesteps. The rewards are defined by the objective of the system, such as completing tasks successfully [Wu *et al.*, 2017; Feng *et al.*, 2016] or negative rewards associated with the cost of human intervention [Rigter *et al.*, 2020; Basich *et al.*, 2020].

Approaches motivated by reducing the amount of human intervention needed typically do not explicitly model the human. Instead, they consider humans as a resource, such as asking them to do tasks the robot cannot do [Rosenthal and Veloso, 2012], or as an example to replicate [Duchetto *et al.*, 2018]. This means that the autonomous system often models the human help as an action choice, which guarantees a deterministic transition to the target state. However, this makes an implicit assumption that humans act perfectly. An SA system that does not consider the weaknesses of the human operator is vulnerable to making sub-optimal choices where the autonomy outperforms the human operator.

The autonomous systems that attempt to create a realistic model of the humans in the environment must consider the probabilities associated with humans making mistakes [Wu *et al.*, 2017; Jean-Baptiste *et al.*, 2015; Charles *et al.*, 2018]. These systems use Markov chains (MCs) to model the stochastic behaviour of humans, such as their likelihood to miss a piece of litter in a picking exercise [Junges *et al.*, 2018]. [Wu *et al.*, 2017] uses a MC to model the impact of the human's fatigue level and their trust in the autonomy on their productivity. This allows them to describe the changes to human behaviour due to internal factors. However, these approaches assume the human state to be fully observable, which is unrealistic in many scenarios. Partially observable Markov decision processes (POMDPs) are used to model humans when the state of the human is not fully observable. Examples of the hidden state space of the human can be their location and speed [Wray *et al.*, 2017], or the goal they are aiming to fulfill [Jean-Baptiste *et al.*, 2015]. We also consider the human state to be hidden, but do so in an SA context. [Javdani *et al.*, 2015] and [Fern *et al.*, 2014] use a POMDP to describe a SA system, where the human's goal is hidden. The human operator's behaviour is dependent on the unknown goal. The agent updates their belief over the true goal by observing the human's actions, and attempts to assist the human. While these papers consider how hidden human variables affect their behaviour in a SA system, unlike our system, they do not consider dynamic hidden human variables that change value during a run, or how the system is affected by human error.

We are interested in modelling the uncertainty in the behaviour of the operator in an SA system. [Wray *et al.*, 2016] considers a similar problem to ours in the domain of semi-autonomous cars. However, they only consider the uncertainty in the engagement level of the human operator during the transfer of control, and otherwise assume humans are perfect drivers. The agent in their problem uses actions such as beeping to gain observations of the human operator. In contrast, SO-SAS infers the state of the human operator by observing the outcomes of their interactions with the environment. This avoids the need to design complicated observation functions for the sensors tracking the human operators.

3 Preliminaries

We propose a new formulation based on a mixed-observability Markov decision processes (MOMDPs) [Ong *et al.*, 2010] to formalise SO-SAS. In a MOMDP, there is an explicit partition of the observable and hidden parts of the state, which allows for better scalability than the corresponding POMDP model.

Definition 1 (MOMDP)**.** A MOMDP is defined by the tuple $\langle S_o, S_h, s_0, b_0, A, \Omega, T, O, R, \gamma \rangle$, where S_o is the set of the observable state factors; S_h is the set of the hidden state factors - the state space of the MOMDP is $S = S_o \times S_h$, and a single state is defined by (s_o, s_h); s_0 is the initial observable state; b_0 is the initial belief distribution of the agent. A belief distribution $b_i(s_h)$ gives the probability of the agent being in a hidden state s_h at time step i; A is the finite set of actions; Ω is the finite set of observations; $T : S \times A \times S \to [0, 1]$ is the transition function, where $T((s_o, s_h), a, (s'_o, s'_h))$ gives the probability of moving to state (s'_o, s'_h) given that action a was taken in state (s_o, s_h); $O : S \times A \times \Omega \to [0, 1]$ is the observation function, where $O((s'_o, s'_h), a, o)$ gives the probability of observing o given action a was taken and the MOMDP transitioned to state (s'_o, s'_h); $R : S \times A \to \mathbb{R}$ is the reward function; and γ is the discount factor.

An MDP is a MOMDP where $S_h = \emptyset$, $\Omega = \emptyset$, and the belief b_0 and observation function O are undefined. We denote MDPs as tuples $\langle S_o, s_0, A, T, R, \gamma \rangle$. An MC is an MDP without action choices. We denote the transition function for an MC as $T : S \times S \to [0, 1]$.

We consider the problem of, given a MOMDP, finding the policy π^* that maximises the infinite-horizon discounted cumulative reward.

4 SA System Modelled With MOMDPs

4.1 Problem Formulation

We consider a set of environment states S^E representing some process we wish to control, along with an initial environment state s^E_0. There are N operators in the SA system that can take control of the process and change its state. The SA agent chooses, at each timestep, which operator will take control of the process. Given a reward function $R^E : S^E \times S^E \to \mathbb{R}$ over environment state transitions, the agent aims to maximise the infinite-horizon discounted cumulative reward collected.

4.2 General Approach

The agent is required to predict which model best describes each operator's behaviour, which evolves stochastically. The agent starts with the initial environment state and an initial belief over how the operators will behave. It then uses this

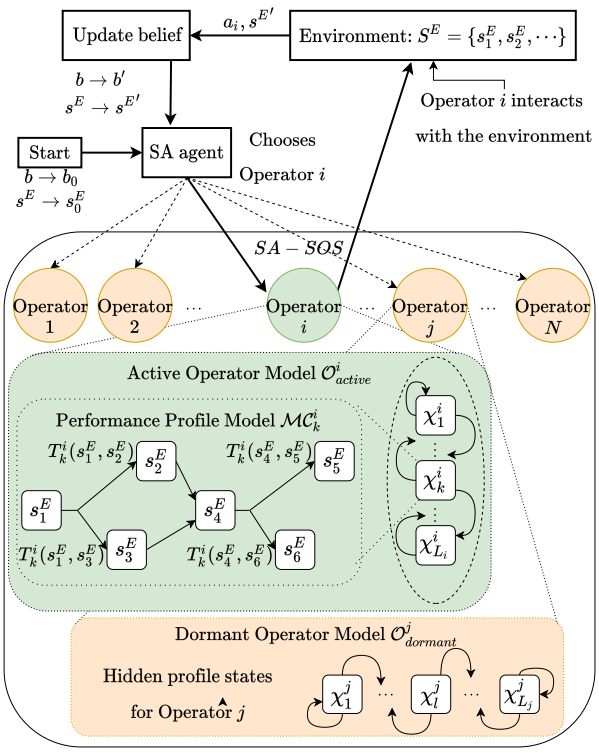

Figure 1: Diagram of the SO-SAS framework.

information to choose an operator. The chosen operator takes control of the system, causing the environment to evolve to a new state. The SA agent then updates its belief by observing the new environment state, and uses it to choose the operator that will take control next. The framework we design for an SA system is shown in Fig. 1. We will formally define each component in the remainder of this section.

Example 1 (UAV surveillance)**.** *We consider an adaptation of the surveillance problem proposed in Feng et al., [2016]. An unmanned aerial vehicle (UAV) is tasked with taking photos at four waypoints. At each waypoint, the UAV will take photos until a "good" photo is taken. The system is comprised of a human operator and autonomous operation. The human might be in a alert state or might be tired, and this influences their ability to take good photos. The aim is to take good photos at all the waypoints while minimising the amount of petrol used. To do so, the SA system should select human control when the human is in an alert state and autonomous control when the human is tired.*

The surveillance problem described above has an environment state space $S^E = S^{Nodes} \times GP_1 \times GP_2 \times GP_3 \times GP_4$. *The set of nodes the UAV could be at is* S^{Nodes}. $GP_i = \{0, 1\}$ *is a binary flag of whether a good photo has been successfully taken at waypoint* W_i, $i \in \{1, 2, 3, 4\}$. *The initial environment state is* $s_0^E = (s^{Nodes} : W_1, GP_1 : 0, GP_2 : 0, GP_3 : 0, GP_4 : 0)$. *If the UAV remains at the same waypoint to take another photo, there is a reward of* -20. *If the UAV transitions from one node to another, there is a reward of* -60. *The states where* $(GP_1 : 1, GP_2 : 1, GP_3 : 1, GP_4 : 1)$ *are absorbing, with zero reward. These rewards reflect the petrol costs incurred by the UAV and the goal state of the mission.*

4.3 Operator Model

Operator $i \in \{1, \ldots, N\}$ can exhibit L_i patterns of behaviour. We model this using *performance profile states*.

Definition 2 (Performance Profile States)**.** The set of performance profile states for operator i is defined as $X^i = \{\chi_1^i, \chi_2^i \cdots \chi_{L_i}^i\}$.

At each timestep the operator is in one performance profile state χ_j^i, and this information is hidden from the SA agent.

Example 2. *In the UAV suveillance problem, the human operator has two performance profile states,* $X^{human} = \{\chi_{alert}^{human}, \chi_{tired}^{human}\}$. *The autonomous operator has one performance profile state* $X^{auto} = \{\chi_1^{auto}\}$.

The behaviour of the ith operator in the jth performance profile state is described by a *performance profile model*. This describes how the ith operator changes the environment state when they are in control and in χ_j^i.

Definition 3 (Performance Profile Model)**.** The performance profile model for performance profile state $\chi_j^i \in X^i$ is an MC $\mathcal{MC}_j^i = \langle S^E, s_0^E, T_j^i \rangle$, where $T_j^i(s^E, s^{E'})$ is the probability of the environment state transitioning from s^E to $s^{E'}$, given that the ith operator is in control and in performance profile state χ_j^i. We assume that the performance profile model accurately represents the operator's behaviour.

Example 3. $\mathcal{MC}_{alert}^{human}$ *and* $\mathcal{MC}_{tired}^{human}$ *are the performance profile models for the human operator, and* $\mathcal{MC}_1^{auto}$ *is the performance profile model for the autonomous operator.* $\mathcal{MC}_{alert}^{human}$, $\mathcal{MC}_{tired}^{human}$ *and* $\mathcal{MC}_1^{auto}$ *have a probability of taking a "good" photo at any waypoint of* 0.9, 0.3 *and* 0.6, *respectively. The transition probabilities in the performance profile models were based on the example given in Feng et al., [2016]. To build a performance profile for a real UAV system, we would gather performance data from the running system, and use this data to build the models for autonomous and human operation.*

At each timestep, the performance profile state of an operator may change. This change is dependent on whether an operator is interacting with the environment or not. Therefore, we model the ith operator interacting with the environment with the *active operator model*, $\mathcal{O}_{active}^i$. When the ith operator is not interacting with the environment, they are described by the *dormant operator model*, $\mathcal{O}_{dormant}^i$.

Definition 4 (Active Operator Model)**.** The active operator model for the ith operator is defined as the MC $\mathcal{O}_{active}^i = \langle S^i, s_0^i, T_{active}^i \rangle$, where $S^i = S^E \times X_i$ is the state space for the ith operator; $s_0^i = (s_0^E, \chi_0^i)$ is the initial state; $T_{active}^i : S^i \times S^i \to [0, 1]$ is the transition function, defined as:

$$T_{active}^i((s^E, \chi^i), (s^{E'}, \chi^{i'})) = \quad T_j^i(s^E, s^{E'}) \cdot P(\chi^{i'} | s^E, \chi^i). \tag{1}$$

$P(\chi^{i'} | s^E, \chi^i)$ is the probability of the ith operator's performance profile state transitioning from χ^i to $\chi^{i'}$, given they are in control and the system is in environment state s^E.

When the ith operator is not interacting with the environment, the change in the performance profile states is described relying only on its their own profile states.

Definition 5 (Dormant Operator Model)**.** The dormant operator model for the ith operator is defined as the MC $\mathcal{O}^i_{dormant} = \langle X^i, \chi^i_0, T^i_{dormant} \rangle$, where $T^i_{dormant}(\chi^i, \chi^{i\prime})$ is the probability of the ith operator's performance profile state transitioning from χ^i to $\chi^{i\prime}$, when they are not in control of the system.

Example 4. *We define the average number of tasks done before a human gets tired to be n_f. We represent this using the human's active operator model, $\mathcal{O}^{human}_{active}$, where we set*

$$\begin{aligned} P(\chi^{human}_{tired}|s^E, \chi^{human}_{alert}) &= 1 - (\frac{1}{2})^{\frac{1}{n_f}}, \\ P(\chi^{human}_{alert}|s^E, \chi^{human}_{alert}) &= (\frac{1}{2})^{\frac{1}{n_f}}, \\ P(\chi^{human}_{tired}|s^E, \chi^{human}_{tired}) &= 1. \end{aligned} \tag{2}$$

As [Feng et al.*, 2016] did not consider the human operator recovering from the tired state, the dormant operator model for the human only contains self-loop transitions. Formally, $T^{human}_{dormant}(\chi^{human}, \chi^{human\prime}) = 1$ if $\chi^{human} = \chi^{human\prime}$, and is 0 otherwise. The value of n_f can be estimated from studies measuring fatigue [Dawson and Reid, 1997].*

4.4 The SO-SAS MOMDP

In SO-SAS, the SA system has N operators that are uncertain and dynamic. We present SO-SAS as a MOMDP, where *the hidden states correspond to the N operators' performance profile states*. The dormant and active operator models for the N operators are used to define the transition functions in the SO-SAS MOMDP. We define the observation space to be the environment states, which are assumed to be fully observable. The observation function is a deterministic Dirac delta function, which returns one if the environment state and the observation are the same, and zero otherwise. Formally:

Definition 6 (SO-SAS)**.** The SO-SAS MOMDP is defined by the tuple $\langle S_o, S_h, s_0, b_0, A, \Omega, T, O, R, \gamma \rangle$, where $S_o = S^E$ is the environment state space; $S_h = X = X^1 \times \cdots \times X^N$ is the hidden performance profile state space for the N operators; $s_0 = s^E_0$ is the initial environment state; b_0 is the initial belief distribution over the performance profile state space X; $A = \{a_1, a_2 \cdots a_N\}$, where a_i denotes the SA agent choosing operator i to take control of the system; $\Omega = S^E$, i.e. the set of observations is the observable state space; $T : (S^E \times X) \times A \times (S^E \times X) \to [0, 1]$ is the transition function. Using the active and dormant operator models for the N operators, we calculate the transition probability for the MOMDP as:

$$\begin{aligned} &T((s^E, \chi^1, \chi^2, \cdots, \chi^N), a_i, (s^{E\prime}, \chi^{1\prime}, \chi^{2\prime}, \cdots, \chi^{N\prime})) = \\ &T^i_{active}((s^E, \chi^i), (s^{E\prime}, \chi^{i\prime})) \times \prod_{\substack{k=1 \\ k \neq i}}^{N} T^k_{dormant}(\chi^k, \chi^{k\prime}); \end{aligned} \tag{3}$$

$O : (S^E \times X) \times A \times \Omega \to [0, 1]$ is the observation function, which we define as:

$$O((s^E, \chi^1, \cdots \chi^N), a, o) = \begin{cases} 1 & \text{if } o = s^E \\ 0 & \text{otherwise;} \end{cases} \tag{4}$$

$R : (S^E \times X) \times A \to \mathbb{R}$ is the reward function obtained from the reward over the environment transitions:

$$\begin{aligned} &R((s^E, \chi), a_i) = \\ &\sum_{(s', \chi') \in S^E \times X} T((s^E, \chi), a_i, (s^{E\prime}, \chi')) R^E(s^E, s^{E\prime}); \end{aligned} \tag{5}$$

and $\gamma \in (0, 1]$ is the discount factor.

Example 5. *The SO-SAS MOMDP for the AUV surveillance problem has $A = \{a_{human}, a_{auto}\}$, allowing the SA agent to choose the controller of the UAV. As there is only one profile state for the autonomous operator, the hidden state space can be simplified to $S_h = X^{human}$. The SA agent initially believes the human to be in the alert profile state, i.e. $b_0(\chi^{human}_{alert}) = 1.0$ and $b_0(\chi^{human}_{tired}) = 0.0$. The discount factor is $\gamma = 1$.*

In the SO-SAS MOMDP, the observation space is simply the environment state. This removes the need to design observation functions for the hidden operator profile states, which can be very complex and problem specific, possibly requiring sensors close to the operator. Observing the transitions between the environment states enables us to refine the belief over the hidden profile states, due to the different models for each performance profile.

5 Experiments

In this section, we apply and evaluate SO-SAS on a simulated domain and a real computer game. We compare SO-SAS to MDP formulations presented in existing work. We also compare robustness to inaccurate modelling for SO-SAS and MDP policies. The MDPs presented in the this section are solved exactly using value iteration. The MOMDPs are solved approximately using a straightforward adaptation of the sampling-based search algorithm POMCP [Silver and Veness, 2010], which extends the definition of history to also include the observable part of the states visited until the current state.

5.1 UAV Surveillance

Domain Description

We applied our approach to the *UAV domain* from Feng *et al.*, [2016]. The UAV flies around a map and takes photos at the waypoints until it has obtained good photos at all four waypoints. There is a human operator and an autonomous operator that can control the UAV, with their models following Examples 1 to 5. In the UAV domain, the average number of tasks done by a human before they enter the tired state is defined as n^d_f.

Methods Compared

We evaluate three types of policies.

SO-SAS policy: The policy found by solving the SO-SAS for each instance using POMCP.

Deterministic Human MDP policy: This makes the assumptions made in Feng *et al.*, [2016]: the performance profile is observable, and the human has a deterministic transition to the tired profile state after n^m_f steps.

Stochastic Human MDP policy: The human is modelled to have a stochastic transition to the tired state. However, unlike

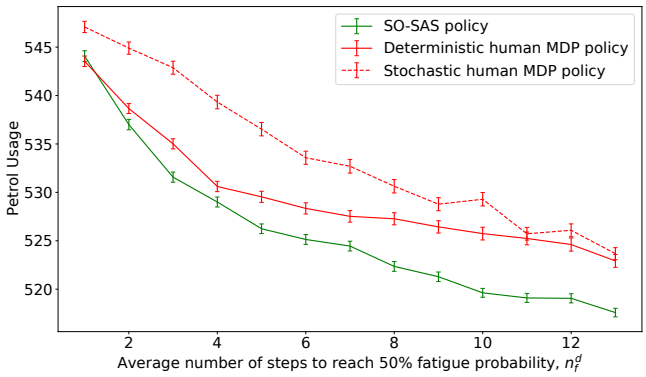

Figure 2: Average petrol usage when following the SO-SAS policy, the deterministic human MDP policy from Feng *et al.*, [2016] and the stochastic human MDP policy on the *UAV domain* over 2000 simulations.

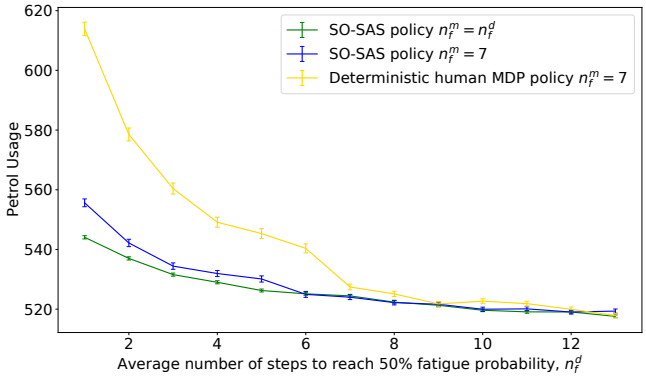

Figure 3: Average Petrol Usage when following the SO-SAS and the deterministic human MDP policy from Feng *et al.*, [2016] where $n_f^m = 7$ in the *UAV domain* over 2000 simulations.

SO-SAS the agent does not hold a belief over the (unobservable) profile state of the human, and therefore samples this state from the model after each action. Sampling is performed according to the probabilities described in Equation 2, and the environment state transitions are not considered.

Results

We applied the these three approaches to the *UAV domain* with a correct model, i.e. with the average number of steps used in the model, n_f^m, set to match the value used in the evaluation environment, n_f^d. To solve the SO-SAS MOMDP, we ran POMCP for 3×10^6 trials. The results are shown in Fig. 2 and show that SO-SAS is able to achieve lower costs than the MDP-based approaches. This is because many times the MDP policies make decisions assuming the human operator to be tired when they are not, and vice-versa. In contrast, by maintaining a belief based on the outcomes of their actions, the SO-SAS policy is able to more accurately predict the human operator profile state, and choose who controls the UAV next accordingly. Given its poor performance, we will not consider the stochastic human MDP policy further in the experiments in order to present clearer plots.

In Fig. 2, the model is assumed to correspond to the domain (i.e. $n_f^m = n_f^d$). However, in practice a human model may not be accurate, e.g. because it is designed to generalise over a range of operators, rather match the current operator. To test the robustness of these policies to inaccurate modelling, we evaluate the SO-SAS policy and the deterministic human MDP policy on the *UAV domain* where the n_f^d value and n_f^m do not align. We use the policies found by solving models with $n_f^m = 7$, and apply the policies to the *UAV domain* where n_f^d ranges from 1 to 13. The results are shown in Fig. 3, where we also plot the value of the SO-SAS policy associated with the correct n_f^m in order to provide a lower bound on the petrol usage for each n_f^d. When $n_f^d < 7$, the petrol used when following the deterministic human MDP policy is much higher than following the SO-SAS policy, because it assumes the human is not tired, but it is likely they are. In contrast, the SO-SAS policy updates the belief distribution to reflect the high failure rate. This allows it to stop granting control to the human operator when they underperform. This allows the performance of the SO-SAS policy to remain closer to optimal.

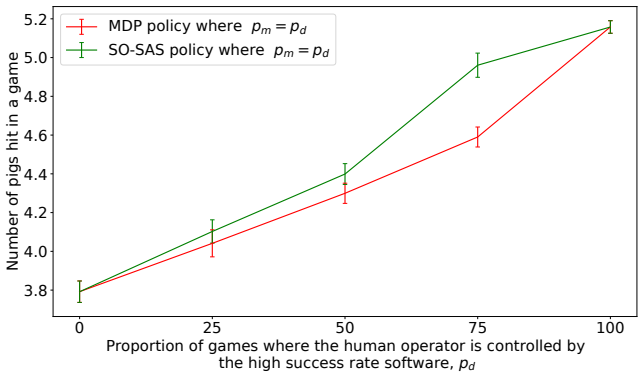

Figure 4: The average number of birds hit in a game when following the SO-SAS policy and the MDP policy, where the domain and the model's p values are aligned ($p_d = p_m$).

5.2 Angry Birds

Domain Description

We considered the game Angry Birds (AB) as an SA system, where two operators can be used to play the game. AB is a puzzle game, where the player uses a catapult to hit pigs hidden in a structure. The player has a fixed number of birds to catapult, and the aim is to maximise the number of pigs hit. Birds are shot sequentially, and each shot is taken by a single player.

Our AB domain has two players, a human and an autonomous operator. We simulate the human operator using two pieces of software, that can play the game through the human interface (i.e. click and drag objects in the game). The two pieces of software have different success rates, to represent human players with low and high skill levels. The software with the lower success rate randomly chooses a pig to shoot at and ignores the presence of obstacles. The software with the higher success rate was developed by Datalabs [Borovička *et al.*, 2014] and won the Angry Birds AI

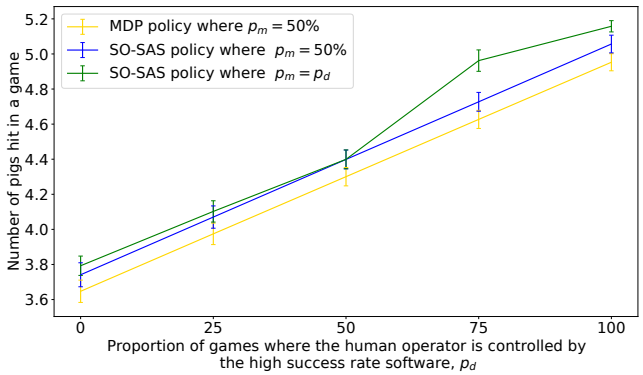

Figure 5: The average number of birds hit in a game when following the SO-SAS policy where the domain and the model's p values are aligned ($p_d = p_m$) and the SO-SAS and MDP policy where the model's p value is $p_d = 50\%$.

competition [Jochen Renz, 2021] in 2013 and 2014. The human is simulated by the high success rate software in $p_d\%$ of games, and the low success software in $(100 - p_d)\%$ of games, where $p_d \in [0, 100]$. The software controlling the human operator does not change during a game. At the start of a game, the agent does not have any information on the current human player's skill level.

The autonomous operator is controlled by a third piece of software that considers all the possible shots to hit the pigs and chooses an unblocked trajectory at random. If there are no clear shots possible, the autonomous operator will choose a random pig to shoot at. Our AB SA evaluation considers a single level with five birds and six pigs.

Methods Compared

We compared the following Angry Birds strategies.

SO-SAS policy: The SO-SAS for the AB game has an environment state space defined by the number of birds and pigs in the environment, and a binary flag of whether the game has ended. At the end of the game, the agent receives a reward equal to the number of pigs hit. There is an additional reward of $+10$ when all of the pigs are hit. At each shot, the agent chooses an action from the set $A = \{a_{auto}, a_{human}\}$.

We construct the performance profile models for the human and the autonomous operator by recording the results of 150 games played by each operator in a known profile state. The transition probabilities for each performance profile model are constructed from the state transition frequencies. These transition probabilities for each automated player are used to generate the associated performance profiles. The human is assumed to have a $p_m\%$ probability of having a high skill level, and a $(100 - p_m)\%$ probability of having a low skill level, where $p^m \in [0, 100]$. The performance profile states for the human operator is $X^h = \{\chi^h_{low}, \chi^h_{high}\}$, and the initial belief is $b_0(\chi^h_{low}) = p_m\%, b_0(\chi^h_{high}) = (100 - p_m)\%$. The SO-SAS model is solved by running POMCP for 3×10^5 trials.

MDP policy: This policy is created from an MDP based on the SA AB domain. As the typical SA formulation does not consider multiple profiles for the human operator, the human operator is represented by a single model. We create the single model using $p_m\%$ of the data from the results of the high skill level human operator, and $(100 - p_m)\%$ of the data from the low skill level human operator.

The MDP agent will choose the operator for the next step based on their model for the default and the mixed human operator. The mixed human operator is represented by a single performance profile, obtained by mixing the two models for the greedy and random performance profiles.

Results

The results from applying the SO-SAS and the MDP policies where $p_m = p_d$ on the domain where p_d ranges from 0% to 100% are shown in Fig. 4. The SO-SAS and MDP policies where $p_m = p_d$ are shown to have similar results in cases where p_d is close to 0 or 100%. In these cases, the agent is certain about the ability of the human operator, so the effect of maintaining a belief distribution over the ability of the human operator is minimal. When p_d is closer to 50%, as there is high uncertainty in the ability of the human, the SO-SAS policy will vary depending on the behaviours exhibited by the human operator. For example, if the human operator misses an easy shot, the agent updates their belief in the human operator having a low ability level. This alters whether the human operator is chosen to take the next shot, and results in the SO-SAS policy outperforming the MDP policy when $p_m = 25, 50, 75\%$.

However, much like the UAV domain, our modelling assumptions may not be accurate. Therefore we tested the robustness of the SO-SAS and MDP policies to inaccurate modelling by evaluating the policies created with $p_m = 50\%$ in the domain where p_d varies between 0 and 100%. This is shown in Fig. 5. The SO-SAS policy outperforms the MDP policy where they model $p_m = 50\%$ in all domains where p_d ranges from 0 to 100%. However, we were unable to observe a significant difference in performance between the SO-SAS policies where $p_m = p_d$ and $p_m = 50\%$.

These results again demonstrate the strength of the SO-SAS approach, compared to an MDP approach, when the performance profile of the operator is not directly observable – an assumption which is likely to hold in reality. The robustness results also demonstrate the importance of the online estimation of operator performance in SA applications. Not only should an SA system not assume that operator performance is observable, but it should also not treat its modelling assumptions as accurate, since human performance cannot be predicted with total accuracy.

6 Conclusion

We have presented a framework for stochastic, dynamic and partially observable operators in a SA system, SO-SAS, using MOMDPs. We compared our SO-SAS policy to MDP policies, and found that our SO-SAS policy significantly outperforms MDP policies.

Acknowledgments

This work was supported by the Defence Science and Technology Laboratory, the EPSRC Programme Grant 'From

Sensing to Collaboration' (EP/V000748/1), the Clarendon Fund at the University of Oxford, and a gift from Amazon Web Services. This document is an overview of UK MOD's Defence Science and Technology Laboratory (DSTL) sponsored research and is released for informational purposes only. The contents of this document should not be interpreted as representing the views of the UK MOD, nor should it be assumed that they reflect any current or future UK MOD policy.

References

[Basich *et al.*, 2020] Connor Basich, Justin Svegliato, Kyle Hollins Wray, Stefan Witwicki, Joydeep Biswas, and Shlomo Zilberstein. Learning to Optimize Autonomy in Competence-Aware Systems. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, pages 123–131, Richland, SC, May 2020. International Foundation for Autonomous Agents and Multiagent Systems.

[Borovička *et al.*, 2014] Tomáš Borovička, Radim Špetlík, and Karel Rymeš. DataLab Birds - Angry Birds AI, August 2014.

[Charles *et al.*, 2018] Jack-Antoine Charles, Caroline P. C. Chanel, Corentin Chauffaut, Pascal Chauvin, and Nicolas Drougard. Human-Agent Interaction Model Learning based on Crowdsourcing. In *Proceedings of the 6th International Conference on Human-Agent Interaction*, HAI '18, pages 20–28, New York, NY, USA, December 2018. Association for Computing Machinery.

[Cubuktepe *et al.*, 2021] Murat Cubuktepe, Nils Jansen, Mohammed Alshiekh, and Ufuk Topcu. Synthesis of Provably Correct Autonomy Protocols for Shared Control. *IEEE Transactions on Automatic Control*, 66(7):3251–3258, July 2021. Conference Name: IEEE Transactions on Automatic Control.

[Dawson and Reid, 1997] Drew Dawson and Kathryn Reid. Fatigue, alcohol and performance impairment. *Nature*, 388(6639):235–235, July 1997. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 6639 Primary_atype: Research Publisher: Nature Publishing Group.

[Duchetto *et al.*, 2018] Francesco Duchetto, Ayse Kucukyilmaz, Luca Iocchi, and Marc Hanheide. Do Not Make the Same Mistakes Again and Again: Learning Local Recovery Policies for Navigation From Human Demonstrations. *IEEE Robotics and Automation Letters*, PP:1–1, July 2018.

[Feng *et al.*, 2016] L. Feng, C. Wiltsche, L. Humphrey, and U. Topcu. Synthesis of Human-in-the-Loop Control Protocols for Autonomous Systems. *IEEE Transactions on Automation Science and Engineering*, 13(2):450–462, April 2016. Conference Name: IEEE Transactions on Automation Science and Engineering.

[Fern *et al.*, 2014] A. Fern, S. Natarajan, K. Judah, and P. Tadepalli. A Decision-Theoretic Model of Assistance. *Journal of Artificial Intelligence Research*, 50:71–104, May 2014.

[Javdani *et al.*, 2015] Shervin Javdani, Siddhartha S. Srinivasa, and J. Andrew Bagnell. Shared Autonomy via Hindsight Optimization. *arXiv:1503.07619 [cs]*, April 2015. arXiv: 1503.07619.

[Jean-Baptiste *et al.*, 2015] Emilie M. D. Jean-Baptiste, Pia Rotshtein, and Martin Russell. POMDP Based Action Planning and Human Error Detection. In Richard Chbeir, Yannis Manolopoulos, Ilias Maglogiannis, and Reda Alhajj, editors, *Artificial Intelligence Applications and Innovations*, IFIP Advances in Information and Communication Technology, pages 250–265, Cham, 2015. Springer International Publishing.

[Jochen Renz, 2021] Jochen Renz. AI Birds.org - Angry Birds AI Competition, 2021.

[Junges *et al.*, 2018] Sebastian Junges, Nils Jansen, Joost-Pieter Katoen, Ufuk Topcu, Ruohan Zhang, and Mary Hayhoe. Model Checking for Safe Navigation Among Humans. In Annabelle McIver and Andras Horvath, editors, *Quantitative Evaluation of Systems*, Lecture Notes in Computer Science, pages 207–222, Cham, 2018. Springer International Publishing.

[Ong *et al.*, 2010] Sylvie C. W. Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. Planning under Uncertainty for Robotic Tasks with Mixed Observability. *The International Journal of Robotics Research*, 29(8):1053–1068, July 2010. Publisher: SAGE Publications Ltd STM.

[Rigter *et al.*, 2020] Marc Rigter, Bruno Lacerda, and Nick Hawes. A Framework for Learning From Demonstration With Minimal Human Effort. *IEEE Robotics and Automation Letters*, 5(2):2023–2030, April 2020.

[Rosenthal and Veloso, 2012] Stephanie Rosenthal and Manuela Veloso. Mobile robot planning to seek help with spatially-situated tasks. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 2067–2073, Toronto, Ontario, Canada, July 2012. AAAI Press.

[Silver and Veness, 2010] David Silver and Joel Veness. Monte-Carlo planning in large POMDPs. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, NIPS'10, pages 2164–2172, Red Hook, NY, USA, December 2010. Curran Associates Inc.

[Wray *et al.*, 2016] Kyle Hollins Wray, Luis Pineda, and Shlomo Zilberstein. Hierarchical approach to transfer of control in semi-autonomous systems. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 517–523, New York, New York, USA, July 2016. AAAI Press.

[Wray *et al.*, 2017] Kyle Hollins Wray, Stefan J. Witwicki, and Shlomo Zilberstein. Online Decision-Making for Scalable Autonomous Systems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 4768–4774, Melbourne, Australia, August 2017. International Joint Conferences on Artificial Intelligence Organization.

[Wu *et al.*, 2017] Bo Wu, Bin Hu, and Hai Lin. Toward efficient manufacturing systems: A trust based human robot collaboration. In *2017 American Control Conference (ACC)*, pages 1536–1541, May 2017. ISSN: 2378-5861.